

CS1010 Programming Methodology

Week 10: Pointers and Functions with Pointer Parameters

To students:

- Some programs for this discussion are on the CS1010 website, under the "Discussion" page.

I. Pointers

Do you really understand how pointers work? For each of the following, trace the code. Line numbers are added for ease of reference.

1. Understanding pointers and addresses.

(a) What is the final value of i?

```
int i = 1;           // Line 1
int *p;              // Line 2

i += 2;              // Line 3
p = &i;              // Line 4
(*p)++;              // Line 5
*p = *p + i;         // Line 6
```

(b) What is the output?

```
int i = 1, j = 1;    // Line 1
int *p, *q;          // Line 2

i += 2;              // Line 3
p = &i;              // Line 4
*p = *p + 3;         // Line 5
q = &i;              // Line 6
i = *p + *q;         // Line 7
printf("i = %d\n", i); // Line 8
q = &j;              // Line 9
i = *p + *q;         // Line 10
printf("i = %d\n", i); // Line 11
p = &j;              // Line 12
i = *p + *q;         // Line 13
printf("i = %d\n", i); // Line 14
```

II. Functions with Pointer Parameters

2. [AY2012/2013 Semester 1 Exam Paper]

Hand trace the following program and write out its output.

```
#include <stdio.h>

int f(int *, int);
int g(int *);

int main(void) {
    int x = 5, y = 5;
    int sum = f(&x, y);
    printf("%d %d %d\n", sum, x, y);
    return 0;
}

int f(int *x, int y) {
    int sum = 0;

    while (*x > 0) {
        sum += g(&y);
        *x -= 1;
    }
    return sum;
}

int g(int *y) {
    *y *= 2;
    return *y;
}
```

3. Lab1 Ex2: Surface area and Longest Diagonal of a Box

In Lab 1 Exercise 2, you wrote two functions **compute_surface_area(int, int, int)** and **compute_diagonal(int, int, int)** to compute the surface area and longest diagonal of a box respectively. The program is available as **box.c** which you may copy from the cs1010 account.

cp ~cs1010/discussion/prog/week9/box.c .

Can you combine the two functions into one, called **compute_surface_area_and_diagonal()**, which passes back both the surface area and length of the longest diagonal?

4. Triangle incenter

In Unit14 exercise #2, you are to compute the centroid of a triangle. Besides the centroid, there are other “centers” of a triangle: circumcenter, orthocenter and incenter. You may refer to

http://jwilson.coe.uga.edu/emat6680/dunbar/assignment4/assignment4_kd.htm

Here, you are to write a program **incenter.c** to compute the incenter of a triangle given its three vertices. Google to search for the formula to compute the coordinates of the incenter. Your program should contain a function **incenter()**. You may use **float** type for all values.

Two sample runs are shown below. The coordinates of the incenter are printed in 2 decimal places.

```
Coordinates of 1st vertex: -1 0
Coordinates of 2nd vertex: 3 0
Coordinates of 3rd vertex: 1 5
Coordinates of incenter = (1.00, 1.35)
```

```
Coordinates of 1st vertex: 63.2 21.8
Coordinates of 2nd vertex: -15 -6
Coordinates of 3rd vertex: -19.2 5.7
Coordinates of incenter = (-11.52, 1.34)
```

III. Design Issues: Programming Methodology and Cohesion

5. After attending CS1010 lecture last week and learning about function with pointer parameters, Brusco is so excited that he replaced this GCD function:

```
// Returns the GCD of a and b
// Precond: a>=0, b>=0 and not both = 0
int brusco_gcd(int a, int b) {
    int remainder;

    while (b != 0) {
        remainder = a % b;
        a = b;
        b = remainder;
    }

    return a;
}
```

with the following function:

```
void brusco_gcd(int a, int b, int *answer) {  
    . . . // body of the function same as above  
  
    *answer = a;  
}
```

He did not make a wise move? Why?

6. After learning in question 5 above that he should stick to his old function instead of using pointer parameter in his GCD function, Brusco, being a very inquisitive and adventurous student (and we all love such students!), tried another new version:

```
void brusco_gcd(int a, int b) {  
    . . .  
  
    printf("The GCD is %d\n", a);  
}
```

His reason being: since the answer (variable a) is to be returned to the caller and get printed anyway, why can't he just save the returning part (and hence make the function a void function) and print the answer inside the function instead?

Comment on his move.

7. **Lab 1 Ex2: Surface area and Longest Diagonal of a Box – Revisit**

In question 3 you attempted to combine the two functions `compute_surface_area()` and `compute_diagonal()` into a single function `compute_surface_area_and_diagonal()`.

Compare the two approaches. Which one do you think is more desirable in terms of good programming methodology?